Assignment 2 - Eitamar Saraf, Moshe Nasletashvili

Details of implementation:

As in the workshop notebook:

Net Architecture:

Conv2d->Conv2d->MaxPool->Dropout->Flatten>Linear>Dropout->Linear->Softmax

We used Cross Entropy Loss and Accuracy as our metric.

Hyperparameters:

batch\_size = 128

#of epochs = 12

learning\_rate =0.1

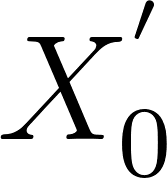
lr\_decay = 1e-6

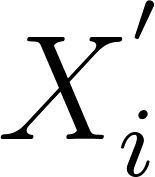
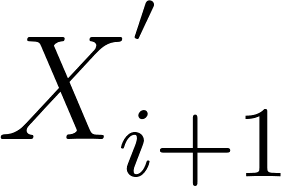
lr\_drop = 20

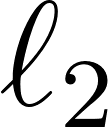
The attack method was FGSM with

Re-Training Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Clean data  accuracy | Attack success rate for | Attack success rate for | Mean L2  perturbation  distance for |
| 1 | 0.975 | 0.97 | 0.93 | 0.29 |
| 2 | 0.976 | 0.93 | 0.90 | 0.24 |
| 3 | 0.977 | 0.93 | 0.88 | 0.29 |
| 4 | 0.978 | 0.89 | 0.82 | 0.29 |
| 5 | 0.978 | 0.85 | 0.81 | 0.29 |

We can see that the attack success on [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X_0%5E'#0) is decreasing as the iterations go on. That is reasonable because we train the model on these adversarial examples. In addition, one can see that the attack success rate on the new adversarial examples randomly generated on each iteration is decreasing. That could make sense because of 2 reasons:

1. The more adversarial data the model “sees”, the more robust it is for the attack.
2. Since we do not exclude [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X_i%5E'#0) from the test set when we continue training to check on [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X_%7Bi%2B1%7D%5E'#0), we are testing on data we’ve already seen (data leakage!).

We believe that reason 2 is more reliable, since we can see that the mean [](https://www.codecogs.com/eqnedit.php?latex=%5Cell_2#0) perturbation stays the same pretty much all over the loop. This means that the difficulty in “tricking” the model is quite stable…

Some adversarial examples:

